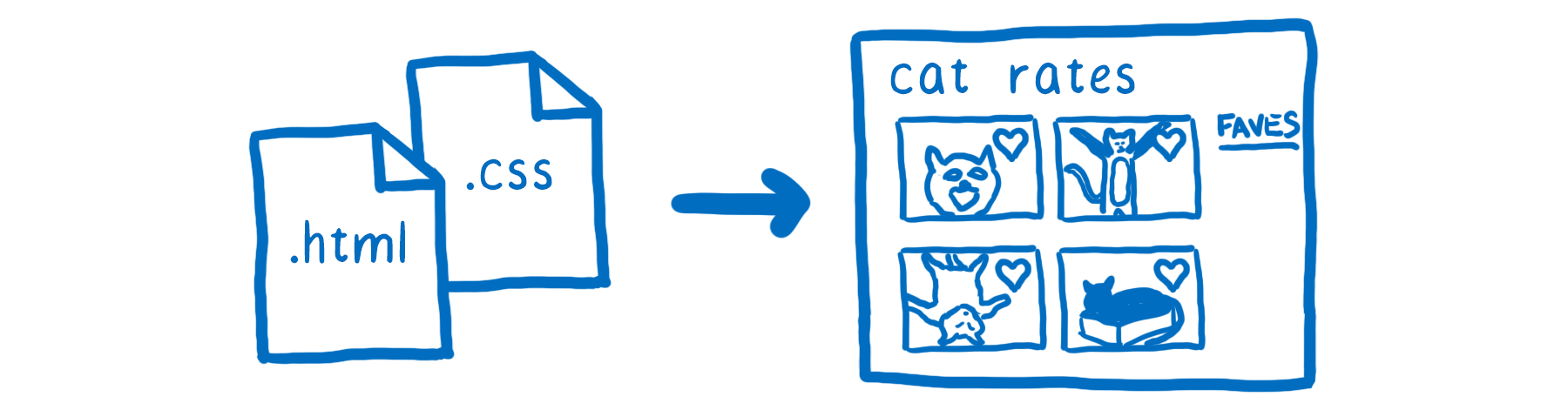
# BeautifulSoup 解析网页: CSS

作者: **莫烦**编辑: **莫烦 2017-12-29**

学习资料:

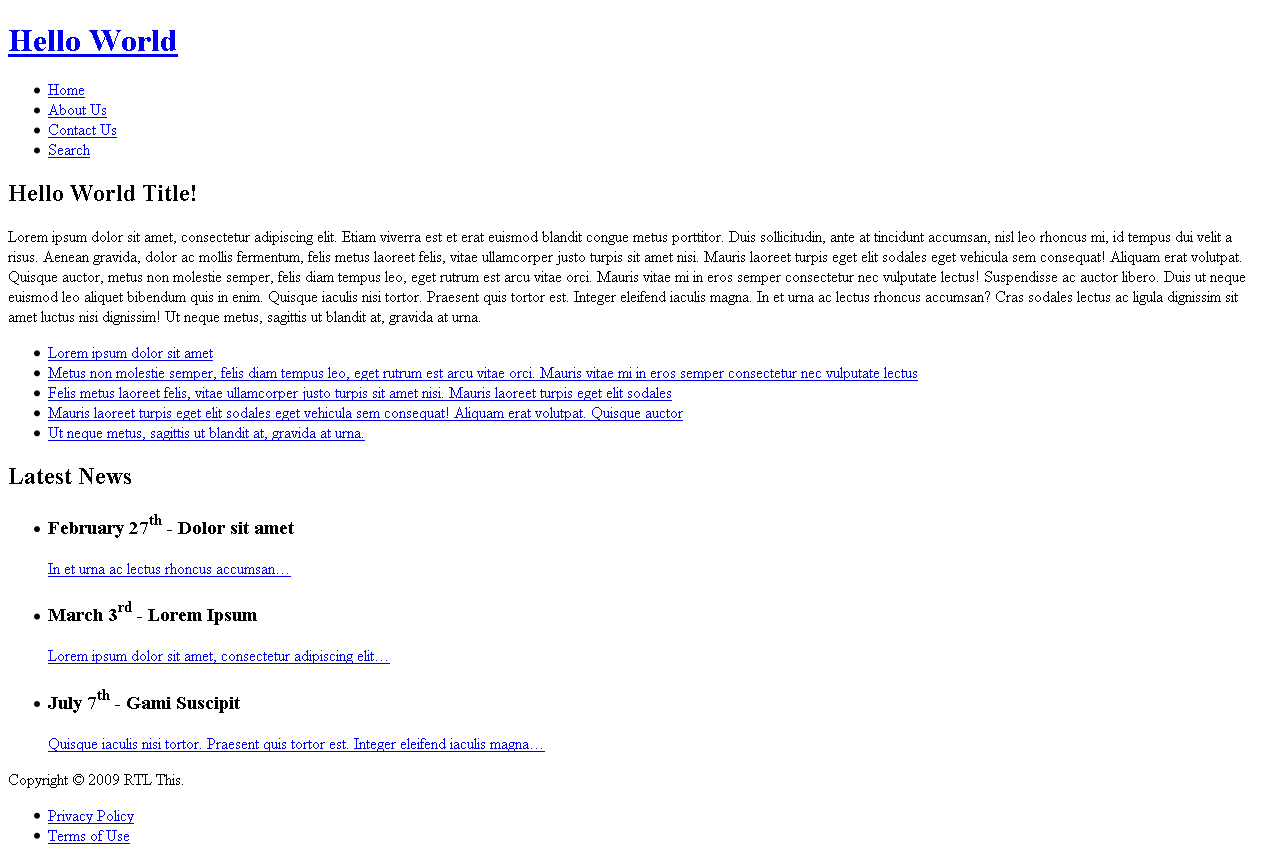
* [本节学习代码](https://github.com/MorvanZhou/easy-scraping-tutorial/blob/master/notebook/2-2-beautifulsoup-css.ipynb)
* BeautifulSoup [英文官网](https://www.crummy.com/software/BeautifulSoup/bs4/doc/), [中文官网](https://www.crummy.com/software/BeautifulSoup/bs4/doc.zh/)
* 本节使用的爬虫[测试网页](https://morvanzhou.github.io/static/scraping/list.html)
* CSS 的[详细规则](https://www.w3schools.com/css/)

BeautifulSoup 十分好用, 能快速定位到你需要的网页信息.[上次](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-01-beautifulsoup-basic/) 我们学着使用了 BeautifulSoup, 这次我们将会了解它更强大的功能, 使用 CSS 的 Class 来选择内容.

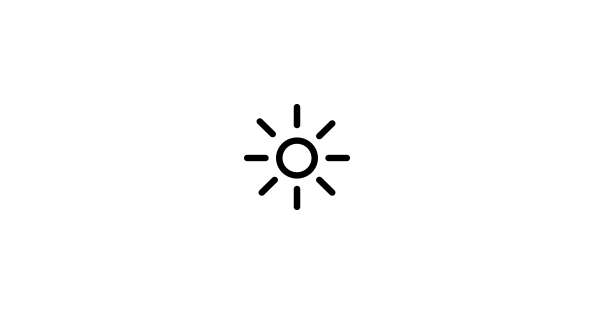
[](https://morvanzhou.github.io/static/results/scraping/2-2-1.png)

## [什么是 CSS](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-02-beautifulsoup-css/#%E4%BB%80%E4%B9%88%E6%98%AF-CSS)

提到这个, 我们肯定需要知道什么是 CSS. HTML 和 CSS 是一对好搭档, 他们共同组成了当今的众多网页. 如果这个世界上没有 CSS, 你看到的所有网页可能都长得像这样. 特别”骨感”!

[](https://morvanzhou.github.io/static/results/scraping/2-2-2.png)

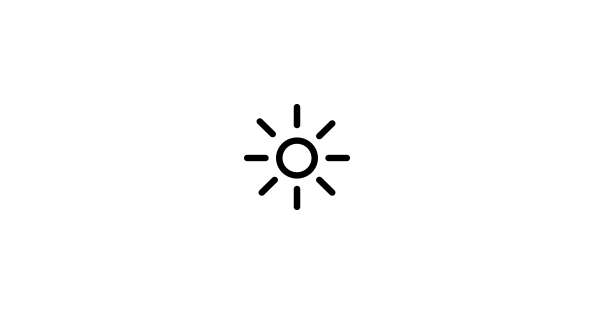
如果有 CSS, 你的网页就变得丰富多彩起来. 文字有了颜色, 字体, 位置也多样了, 图片也有规则了.

[](https://morvanzhou.github.io/static/results/scraping/2-2-3.png)

所以, CSS 主要用途就是装饰你 “骨感” HTML 页面. 如果将 HTML 比喻成没穿衣服的人, 那 CSS 就是五颜六色的衣服. 穿在人身上让人有了气质. CSS 的规则很多, 好在如果你只是需要爬网页, 你并不需要学习 CSS 的这些用法或规则, (如果你想, 你可以看到[这里](https://www.w3schools.com/css/)), 你只需要注意 CSS 的一条规则就能玩转爬虫了.

## [CSS 的 Class](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-02-beautifulsoup-css/#CSS-%E7%9A%84-Class)

这条规则就是 CSS 的 Class, CSS 在装饰每一个网页部件的时候, 都会给它一个名字. 而且一个类型的部件, 名字都可以一样. 比如我们这个[练习网页](https://morvanzhou.github.io/static/scraping/list.html). 里面的字体/背景颜色, 字体大小, 都是由 CSS 来掌控的.

[](https://morvanzhou.github.io/static/results/scraping/2-2-4.png)

而 CSS 的代码, 可能就会放在这个网页的 <head> 中. 我们先使用 Python 读取这个页面.

**from** bs4 **import** BeautifulSoup

**from** urllib.request **import** urlopen

# if has Chinese, apply decode()

html = urlopen("https://morvanzhou.github.io/static/scraping/list.html").read().decode('utf-8')

print(html)

在 <head> 中, 你会发现有这样一些东西被放在 <style> 里面, 这些东西都是某些 class 的 CSS 代码. 比如 jan 就是一个 class. jan 这个类掌控了这个类型的背景颜色. 所以在 <ul class="jan"> 这里, 这个 ul 的背景颜色就是黄色的. 而如果是 month 这个类, 它们的字体颜色就是红色.

<head>

...

<style>

.jan {

background-color: yellow;

}

...

.month {

color: red;

}

</style>

</head>

<body>

...

<ul>

<li class="month">一月</li>

<ul class="jan">

<li>一月一号</li>

<li>一月二号</li>

<li>一月三号</li>

</ul>

...

</ul>

</body>

这样, 我们就知道, 有时候, 网页中, 这种 class 归类一些组件还是很有用的. 比如我就想找 jan 下面的这些 <li>. 我就能通过寻找 class="jan" 找到它们. BeautifulSoup 就能这么干.

## [按 Class 匹配](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-02-beautifulsoup-css/#%E6%8C%89-Class-%E5%8C%B9%E9%85%8D)

按 Class 匹配很简单. 比如我要找所有 class=month 的信息. 并打印出它们的 tag 内文字.

soup = BeautifulSoup(html, features='lxml')

# use class to narrow search

month = soup.find\_all('li', {"class": "month"})

for m in month:

print(m.get\_text())

"""

一月

二月

三月

四月

五月

"""

或者找到 class=jan 的信息. 然后在 <ul> 下面继续找 <ul> 内部的 <li> 信息. 这样一层层嵌套的信息, 非常容易找到.

jan = soup.find('ul', {"class": 'jan'})

d\_jan = jan.find\_all('li') # use jan as a parent

for d in d\_jan:

print(d.get\_text())

"""

一月一号

一月二号

一月三号

"""

如果想要找到一些有着一定格式的信息, 比如使用正则表达来寻找相类似的信息, 我们在 BeautifulSoup 中也能嵌入正则表达式, 让 BeautifulSoup 更为强大. 怎么用, 我们就接着往下看啦.